ELSEVIER

Theoretical Computer Science 210 (1999) 99–120

Theoretical Computer Science

Relatively recursive reals and real functions[1]

Chun-Kuen Ho *

DSO National Laboratories, 20 Science Park Drive, Singapore 118230, Singapore

Abstract

Intuitively, a real number is recursive if we can get as accurate an approximation as we like, using a mechanical procedure such as a Turing machine. A real function is recursive if its value at a point x in its domain can be approximated effectively given an approximation to x. However, since there are only countably many Turing machines, there must be uncountably many non-recursive reals and functions. In this paper, we study some of these non-recursive reals and functions using a more recursion theoretic approach, via the degree of unsolvability. In particular, we are interested in reals and real functions that are *relatively recursive in* $\emptyset'$, where $\emptyset'$ is the jump of the recursive degree $\emptyset$. Inspired by the Shoenfield Limit Lemma, we show that a real is $\emptyset'$-recursive if and only if it is the limit of a recursive sequence of rationals. We then give three characterizations of a $\emptyset'$-recursive function which are stated in terms of Turing machine, uniform convergence, and sequential computability with uniform continuity. A proof of their equivalence using a finite injury priority argument is given. With the new definitions, we can now give an upper bound to the difficulty of some uncomputable analysis operator such as differentiations and root findings.

Keywords: Relative recursiveness; Recursive in $\emptyset$; Uniformly limiting recursive

1. Introduction

In [19], Turing first introduced the concept of a recursive real number (we use the terms "recursive", "computable", "effective" and "decidable" interchangeably, assuming Church's thesis). Intuitively, a real number is recursive if we can effectively generate its decimal (or binary) expansion as long as we wish, thus we can get as accurate as we like. A few different definitions of recursive real numbers have been found to describe the same class of reals (see [14]). However, they are not equivalent in the sense that a description of such a real number with one definition may not be effectively

* Email: hchunkue@dso.org.sg.

[1] This paper was first written as a technical report [5] when the author was a graduate student at the University of Chicago.

0304-3975/99/$ – see front matter
PII: S0304-3975(98)00098-X

translated to a description in another definition. The most useful and convenient one, the Cauchy sequence definition, states that we can effectively find as close a rational approximation as we wish. In the 1950s, Grzegorczyk, Lacombe and others studied the notion of recursive real functions. Intuitively, a real function is recursive if there is a uniform and effective way (for example, using a Turing machine) to approximate its value at a point x, if an approximation to x itself is given as input. Again, several different definitions are formulated and shown to be equivalent [3]. More recently, Pour-El and Richards [13] study the computability issue in analysis and physics, using an axiomatic approach. They define a "computability structure" in a Banach space to be the set of all sequences of elements in the Banach space satisfying some axioms. They then study the effect of various kind of linear operators between Banach spaces on the computability property of the elements. On the other hand, Ko [9] investigates on the computational complexity aspect on reals and real functions and how the complexity of real analysis relates to problems in classical discrete complexity classes. Here we take a different approach to recursive analysis; we define a "relatively" recursive real number and real function, with respect to a given oracle. Since the oracle itself may well be non-recursive, the definition thus includes many more non-recursive reals and real functions. We try to characterize these functions via the degree of unsolvability as in classical recursion theory.

Similar to a recursive real, a real number is recursive relative to an oracle B if we can effectively, using the oracle, generate as long a decimal expansion as we wish. B-recursive real functions can be similarly defined. In particular, we are most interested in $\emptyset'$-recursive recursive reals and real functions, where $\emptyset'$ is the jump of the recursive Turing degree $\emptyset$, and is Turing equivalent to the halting set K. Motivated by the Limit Lemma in recursion theory, we show that a real number is $\emptyset'$-recursive if and only if it is the limit of a recursive sequence of rationals. Next we formulate three different definitions of $\emptyset'$-recursive real functions which are shown to be equivalent on a bounded domain. The first is based on Turing machine as discussed above. The second definition, which is also termed "uniformly limiting recursive", states a real function is $\emptyset'$-recursive if it is the limit of a uniformly converging sequence of recursive functions. The third one, similar to one of the original definitions of Grzegorczyk, says a real function is $\emptyset'$-recursive if (1) it maps a recursive sequence of reals into a $\emptyset'$-recursive sequence of reals, and (2) it must be uniformly continuous and the modulus is recursive in $\emptyset'$. The proof of equivalence between the first and the second, which uses a finite injury priority argument, forms a major part of this paper. The equivalence of the definitions, which are stated in very different terms, leads us to believe that our definition of a $\emptyset'$-recursive real function is correct.

Next we show some applications of the definition of relative recursiveness. We know that even if a recursive function has roots on a bounded domain, none of the roots need to be recursive. However, we show that it must have a $\emptyset'$-recursive root and give a $\emptyset'$-recursive algorithm to find one such root. Also, while integration preserves recursiveness, it is known that the derivative of a continuously differentiable recursive function need not be recursive. We show that this derivative has to be $\emptyset'$-recursive.

2. Preliminaries

Let $\mathbb{N}$ be the set of all natural numbers (including zero), $\mathbb{Q}$ be the set of all rational numbers, $\mathbb{R}$ be the set of all real numbers, and $\mathbb{D}$ be the set of all dyadic rational numbers, i.e., all rationals in the form of $\pm m/2^n$ where $m,n \in \mathbb{N}$. Let $\mathbb{D}_n$ be the set of all dyadic rationals with denominators 2^n. Hence $\mathbb{D} = \bigcup_{n\in\mathbb{N}} \mathbb{D}_n$. Also let $\mathbb{D}_n^{[a,b]}$ represent $\mathbb{D}_n \cap [a,b]$. Note that $\mathbb{D}_n^{[a,b]}$ is thus finite for any a and b. Since a rational number can be represented by two integers, we sometime, without loss of generality, assume that the input or output of a Turing machine to be rational numbers. The context will make it clear whether the input/output is an integer or a rational. Also, the term "recursive" may be used to describe a real number, a real function, a set of integers, an integer function or an algorithm. Again the context will make it clear what object we are describing. Furthermore the term "recursive", when used on a set of integers, an integer function or an algorithm, has the same meaning as in classical recursion theory.

Definition. A sequence of rational numbers $\{r_n\}$ is *recursive* if there exists a Turing machine M such that on input n, M halts and gives output r_n. In notation, $M(n) = r_n$. Double or n-tuple recursive sequences can be similarly defined.

Definition. A real number x is *recursive* if there is a recursive sequence $\{r_n\}$ of rational numbers and a recursive modulus function $d : \mathbb{N} \to \mathbb{N}$ such that $n > d(n_0)$ implies $|r_n - x| \leqslant 2^{-n_0}$. We also say this recursive sequence *converges effectively* to x, or, the Turing machine producing this sequence *computes x effectively*. This is known as the Cauchy sequence definition.

Definition. A sequence of real numbers $\{x_n\}$ is *recursive* if there is a recursive double sequence $\{r_{nk}\}$ of rationals such that $r_{nk} \to x_n$ as $k \to \infty$, effectively in n and k. Or, equivalently, there exists a Turing machine M such that $M(n,\cdot)$ computes x_n effectively.

The results stated in this section are well known in recursive analysis. For a good reference, one can refer to Pour-El and Richards [13] or Ko [9].

Proposition 1. *A real number x is recursive if and only if there is a recursive sequence of rationals $\{r_n\}$ such that for all n, $|r_n - x| \leqslant 2^{-n}$.*

The above proposition gives a more convenient sequence to work with since we do not have to worry about the modulus function. It says that we can convert any recursive modulus function into the identity modulus function.

As in the case of a usual real number, a recursive real number can also be defined via Dedekind cut, binary digit expansion or nested closed intervals. However, we will not go into detail here since we will not be using these definitions. Rice [14] provides a detail treatment for such constructions.

An important property of recursive real numbers is that we cannot effectively check their equality. A proof of this can be found in [14].

Proposition 2. *It is undecidable that whether two recursive reals x and y are equal. However, if we are given that $x \neq y$, then there is an effective procedure to determine whether $x > y$ or $x < y$.*

We need the definition of an oracle Turing machine before we turn to the definition of a real-valued function on $\mathbb{R}$.

Definition. An oracle Turing machine M with an oracle B is a Turing machine with one extra tape, the query tape, and two special states, the query state and the answer state. When making a query, M first puts the argument n on the query tape and then enters the query state. Then the oracle will take over and in one step puts the answer $B(n)$ on the query tape, erasing the original argument and then enter the answer state. We denote the output (if it ever halts) from an oracle Turing machine with oracle B on input n to be $M^B(n)$. A multiple-oracle Turing machine can be similarly defined.

In this paper, the oracle B will sometime be a sequence $\{r_n\}$. Hence the value $B(n)$ actually means r_n. In other occasion, B would be a set of integers. In this case, we mean that the oracle function is the characteristic function of B.

Let CF_x (for Cauchy function, as in [9]) denote the set of dyadic rational sequences ϕ (not necessarily recursive) converging to x with the identity modulus function, i.e., $\phi(n) \in \mathbb{D}_n$ and $|\phi(n) - x| \leqslant 2^{-n}$ for all $n \in \mathbb{N}$, where $\phi(n)$ denotes the nth term in ϕ. There is one special Cauchy function in CF_x for every x. We call this function β_x, the *standard Cauchy function* for x and it has the property that $|\beta_x(n)| \leqslant |x| < |\beta_x(n)| + 2^{-n}$ for all $n \in \mathbb{N}$. This $\beta_x(n)$ is the binary expansion of x with n binary places. Note that this binary expansion is unique for any non-dyadic rational. For a positive dyadic rational $d \in \mathbb{D}_k$ such that $d = m/2^{-k}$ for some odd positive m, we can define $\beta_d(i) = [m/2^{i-k}]/2^i$ where $[y]$ means the biggest integer less than or equal to y. A similar construct exists for a negative dyadic rational (the number zero is trivial).

Definition. A function $f : \mathbb{R} \to \mathbb{R}$ is *recursive* if and only if there exists an oracle Turing machine M such that $(\forall x \in \mathbb{R})(\forall \phi \in CF_x)\, |M^\phi(n) - f(x)| \leqslant 2^{-n}$ for all $n \in \mathbb{N}$.

In other words, from the output accuracy requirement, M computes the required input accuracy and asks the oracle for this required approximation to x, and then computes an approximation to $f(x)$ within an error of 2^{-n}. Since any computation by the machine can only make a finite number of queries (thus finitely much amount of information of the oracle ϕ is used), the function computed must be continuous. An intuitive reason is, if a function is not continuous, then, by Proposition 2, for an input x close to the break point, we cannot decide in any finite amount of time whether this x is indeed the break point. Thus we cannot approximate the value of the function at the break point. Continuous functions do not have this problem (see Lemma 15).

Definition. A sequence of functions $\{f_n\}$ is *recursive* if there exists an oracle Turing machine M such that $M(n, \cdot)$ is an oracle Turing machine that computes f_n.

We are mostly interested in functions on a closed interval $[a, b]$ with recursive end-points a and b. However, without loss of generality, $[0, 1]$ can be used instead of an arbitrary $[a, b]$. On such a closed interval, another characterization of a recursive function can be given. Note that a sequence of functions can also be similarly characterized by appropriate indexing.

Theorem 3. *A function $f : [0, 1] \to \mathbb{R}$ is recursive if and only if*
(1) *for any recursive sequence $\{x_k\}$ of reals, $\{f(x_k)\}$ is also recursive, and*
(2) *f is effectively uniformly continuous, i.e., there exists a recursive function d : $\mathbb{N} \to \mathbb{N}$ such that $\forall k \in \mathbb{N}$, $\forall x, y \in [0, 1]$, $|x-y| \leqslant 2^{-d(k)}$ implies $|f(x)-f(y)| \leqslant 2^{-k}$.*

This definition is due to Grzegorczyk and Lacombe [3], while the proof of the equivalence can be found in [9].

An early important property of recursive functions is that they are closed under effective and uniform convergence.

Theorem 4. *If a recursive sequence of functions on $[0, 1]$ converges uniformly and effectively (i.e, the modulus function for the uniform convergence is recursive) to a function f, then f is recursive on $[0, 1]$.*

Weierstrass's approximation theorem can also be effectivized and used as an alternative definition of a recursive function. This definition is due to Caldwell and Pour-El [1].

Theorem 5 (Effective Weierstrass approximation). *A function f on $[0, 1]$ is recursive if and only if there exists a recursive sequence of polynomials on $[0, 1]$ such that it converges effectively and uniformly to f.*

The proofs for the above can be found in [13].

3. Relatively recursive reals

Motivated by the arithmetic hierarchy in recursion theory, we look at reals that can be computed with respect to a given oracle.

Definition. A real number x is *B-recursive* if there exists an oracle Turing machine M with oracle B such that $|M^B(n) - x| \leqslant 2^{-n}$ for all $n \in \mathbb{N}$. Sometime we also say that x is *recursive in* B. A sequence of B-recursive real numbers can be similarly defined.

The idea is to consider what reals are now computable, with the help of the extra oracle. It is obvious that a recursive real is also B-recursive for any oracle B. There is one oracle that we are most interested in: the halting set K; the set of indices of all Turing machines that halt on an empty input tape, under a fixed and acceptable numbering. This K is recursively enumerable (r.e.) but not recursive (as a subset of $\mathbb{N}$) and is Turing equivalent to $\emptyset'$, the jump of the recursive degree $\emptyset$. The most obvious example of a $\emptyset'$-recursive real number is $x = \sum_{n \in K} 2^{-n}$. This number x is non-recursive because if it were, we can use the Turing machine for x to compute an approximation to x within an error of $2^{-(n+1)}$ and then we can tell whether $n \in K$ (assuming we already know about the status of $0, \dots, n-1$, by induction hypothesis). Thus K would be decidable, a contradiction.

Specker [18] has constructed a recursive function on $[0,1]$ such that it has roots but none of the roots is recursive. [2] Any such function must have infinitely many roots as it can be shown that an isolated root of a recursive function must be recursive. It is also known that there does not exist an effective procedure to find a root of a recursive function even if it is known to have recursive roots [13]. In the following, however, we show that by using the oracle $\emptyset'$, we can find a root effectively, and thus that root is $\emptyset'$-recursive. (Obvious not all roots are $\emptyset'$-recursive; consider the function $f \equiv 0$.)

Lemma 6. *If f is recursive on* $[0,1]$, *then the following are equivalent*:
(1) *f has a root*; *and*
(2) $(\forall n \in \mathbb{N})(\exists r \in \mathbb{D}^{[0,1]}_{d(n)})[|M^{\beta_r}(n)| \leqslant 2^{-n+1}]$ *where M is the Turing machine computing f and d is the recursive modulus function for uniform continuity of f.*

Proof. ((1) $\Rightarrow$ (2)): Suppose x is a root for f, i.e., $f(x)=0$. Then $(\forall n)\,|f(x)| \leqslant 2^{-n}$ and since d is the modulus for the uniform continuity for f, we have $(\forall n)(\forall y \in [0,1])$,

$$|y-x| \leqslant 2^{-d(n)} \Rightarrow |f(y)| \leqslant |f(y)-f(x)| + |f(x)| = |f(y)-f(x)| \leqslant 2^{-n}.$$

Hence, we have

$$(\forall n)(\exists r \in \mathbb{D}^{[0,1]}_{d(n)})\,|f(r)| \leqslant 2^{-n}.$$

Now M is the Turing machine computing f, therefore $(\forall n)(\exists r \in \mathbb{D}^{[0,1]}_{d(n)})$,

$$|M^{\beta_r}(n)| \leqslant |M^{\beta_r}(n) - f(r)| + |f(r)| \leqslant 2^{-n} + 2^{-n} = 2^{-n+1}.$$

((2) $\Rightarrow$ (1)): Suppose we have

$$(\forall n)(\exists r \in \mathbb{D}^{[0,1]}_{d(n)})[|M^{\beta_r}(n)| \leqslant 2^{-n+1}].$$

[2] Actually, Specker shown that there exists a recursive function f that does not attain its maximum at any recursive points. But we can convert this f to $f(x)-m$ where m is the maximum value of f on $[0,1]$, so that the maximum points are also the roots. This maximum value m is recursive, see [13].

Therefore, in particular, we have $(\forall n)(\exists r \in [0,1])$ such that

$$|f(r)| \leqslant |f(r) - M^{\beta_r}(n)| + |M^{\beta_r}(n)| \leqslant 2^{-n} + 2^{-n+1} = 3 \cdot 2^{-n}.$$

Let $A_n = \{x \in [0,1] : |f(x)| \leqslant 3 \cdot 2^{-n}\}$. So A_n is not empty and $A_{n+1} \subset A_n$ for all n. Note that A_n are closed because f is continuous. Hence $\bigcap_n A_n$ is not empty and let $x_0 \in \bigcap_n A_n$. We have $|f(x_0)| \leqslant 3 \cdot 2^{-n}$ for all n which implies $f(x_0) = 0$, i.e., x_0 is a root of f in $[0,1]$. □

Theorem 7. [3] *If f is recursive on $[0,1]$, then we can tell whether f has a root in $[0,1]$ by using the $\emptyset'$ oracle. Furthermore, if f has a root, then it must have a $\emptyset'$-recursive root and we can find one such root effectively in $\emptyset'$.*

Proof. It suffices to check statement (2) of the previous lemma to tell whether f has a root in $[0,1]$. But that statement is a Π^0_1 statement because the existential quantifier in it is bounded. Hence we can use the $\emptyset'$ oracle to answer whether f has a root in $[0,1]$. If so, to find such a root, divide $[0,1]$ into $[0,\frac{1}{2}]$ and $[\frac{1}{2},1]$ and ask $\emptyset'$ whether f has a root in each half. Let $[a_1,b_1] = [0,\frac{1}{2}]$ or $[\frac{1}{2},1]$ accordingly and repeat the process on $[a_1,b_1]$. We will therefore have a $\emptyset'$-recursive sequence of nested intervals $[a_n,b_n]$ such that the width tends to 0 effectively. The unique point $x \in \bigcap_n [a_n,b_n]$ is thus a $\emptyset'$-recursive root of f. □

Next we investigate the property on the limits of recursive sequences of rationals. This is motivated by the Shoenfield Limit Lemma (see, for example, [17]) in recursion theory. We state a form that is suitable for our use here (note that all functions in the lemma are on integers):

Lemma 8 (Shoenfield limit). *For any function f, $f \leqslant_T \emptyset'$ if and only if there is a recursive sequence of functions $\{f_s\}$ such that $f = \lim_s f_s$.*

Note that in the above, $f = \lim_s f_s$ means that for any $k \in \mathbb{N}$, $f(k) = f_s(k)$ for all but finitely many s. Next we specify some convention on the notations used. Let $\{K_s\}$ be a recursive enumeration of the halting set K such that K_s are increasing with $K_{s+1} - K_s$ having exactly one element, and $K = \bigcup_s K_s$. [4] The notation M_s denotes a Turing machine M running for s steps (no matter whether it has produced any output by s steps, M_s stops running after s steps.) An up arrow after a Turing machine indicates that it does not halt (*diverges* or *is undefined*), and a down arrow indicates that it halts. For instance, $M_s(n)\downarrow = y$ says the Turing machine M on input n halts in s steps and produces an output y; $M(n)\uparrow$ indicates it does not halt on input n, i.e., $M_s(n)\uparrow$ for all s. Note that if $M_s(n)\downarrow = y$ then $M_t(n)\downarrow = y$ for all $t \geqslant s$ and they must

[3] Kurtz [10] pointed out if such an f has a root, then it must have a "low" root. See the Low Basic Theorem in Soare [17]. Also, although f must have a $\emptyset'$-recursive root, the question whether this root is in fact recursive or not, is undecidable in $\emptyset'$.

[4] Construction of such an enumeration can be found in [17].

produce the same value. For an oracle Turing Machine M^K with K as the oracle, we can use $M_s^{K_s}$ as an approximation. In this case, we can only be sure that for all n, $\exists s_0 \forall s\,(s \geqslant s_0 \Rightarrow M_s^{K_s}(n)\downarrow = M^K(n))$.

Theorem 9. *A real number x is $\emptyset'$-recursive if and only if it is the limit of a recursive sequence of rationals.*

Proof. ($\Rightarrow$) Suppose x is $\emptyset'$-recursive, i.e., there exists an oracle Turing machine M with oracle K such that

$$|M^K(e) - x| \leqslant 2^{-e} \quad \text{for all } e \in \mathbb{N}.$$

We will construct a recursive sequence of rationals $\{r_n\}$ such that all of the following *requirements* R_e for all $e \geqslant 1$ are satisfied:

$$R_e : (\exists s_e)(\forall n \geqslant s_e)\,[|r_n - x| \leqslant 2^{-(e-2)}],$$

which implies that $r_n \to x$.

Construction:
Stage $s = 0$. Let $r_0 = 0$.
Stage $s+1$. For all $e \leqslant s+1$, if $M_{s+1}^{K_{s+1}}(e)\uparrow$ or $M_{s+1}^{K_{s+1}}(e)\downarrow \neq M_s^{K_s}(e)\downarrow$ then for all $i \geqslant e$, requirement R_i is said to be *injured* and *requires attention.*

Choose least $e \leqslant s+1$ such that R_e requires attention. If $M_{s+1}^{K_{s+1}}(e)\downarrow$ and for all $e' < e$,

$$|M_{s+1}^{K_{s+1}}(e) - M_{s+1}^{K_{s+1}}(e')| \leqslant 2^{-e} + 2^{-e'}$$

then let $r_{s+1} = M_{s+1}^{K_{s+1}}(e)$. The requirement R_e is said to have *received attention.* If there is no such e, let $r_{s+1} = r_s$.

Claim. *For all e, R_e is injured finitely often and will eventually receive attention for the last time and not be injured anymore.*

Proof. We will prove this by induction. For the case $e = 1$, R_1 will not be injured by any other requirements. Let s_0 be a stage such that $(\forall s \geqslant s_0)\,M_s^{K_s}(1)\downarrow = M^K(1)$. If R_1 has not received attention by then, it will receive and will not be injured anymore. Suppose by induction hypothesis the claim is true on $R_{e'}$ for all $e' < e$. This means there is a stage t_0 such that for all $t \geqslant t_0$ all computations $M_t^{K_t}(e')\downarrow = M^K(e')$. Now let $s_e \geqslant t_0$ be a stage such that for all $s \geqslant s_0$, $M_s^{K_s}(e)\downarrow = M^K(e)$. If R_e has not received attention by then, it will require and receive attention for the last time because for all $e' < e$,

$$\begin{aligned} |M_{s_e}^{K_{s_e}}(e) - M_{s_e}^{K_{s_e}}(e')| &= |M^K(e) - M^K(e')| \\ &\leqslant |M^K(e) - x| + |x - M^K(e')| \leqslant 2^{-e} + 2^{-e'}. \end{aligned}$$

Claim. For all e, R_e is satisfied.

Proof. Let s_e be the stage where R_e receives attention for the last time. For any $n \geqslant s_e$, by construction $r_n = M_t^{K_t}(i)$ for some $i \geqslant e$ and some t such that $s_e \leqslant t \leqslant n$. Hence

$$\begin{aligned}|r_n - x| = |M_t^{K_t}(i) - x| &\leqslant |M_t^{K_t}(i) - M_t^{K_t}(e)| + |M_t^{K_t}(e) - x| \\ &\leqslant 2^{-i} + 2^{-e} + 2^{-e} \leqslant 2^{-(e-2)}\end{aligned}$$

by construction.

($\Leftarrow$) Suppose we are given a recursive sequence $\{r_n\}$ converging to x. We construct a sequence $\{q_s\}$ recursive in $\emptyset'$ as follow: Set $n_{-1} = 0$. For $s \geqslant 0$, define

$$n_s = (\mu n \geqslant n_{s-1})[(\forall i, j \geqslant n)\,|r_i - r_j| \leqslant 2^{-s}].$$

Let $q_s = r_{n_s}$. The expression μn in the above construction means "the least n such that ...". Such a least n exists because $\{r_n\}$ converges and we can use the oracle $\emptyset'$ to find this least n. Hence, the sequence $\{q_s\}$ is recursive in $\emptyset'$ and clearly $|q_s - x| \leqslant 2^{-s}$. □

In the forward direction of the above proof, we have used the *finite injury priority method* where a complex statement to be proved is broken down into a countably infinite list of *requirements*. We then attempt to satisfy the requirement one by one, according to some *priority*. In the course of doing so, some previously satisfied requirements may be *injured* by a newly developed event or simply because another requirement of higher priority is also injured. However, we will be able to show that all requirements can only be injured *finitely* often and hence will eventually be satisfied. For more discussion of the priority method, one can refer to [17]. In Theorem 16 and [7], this method will once again be used but in a more complex setting.

The set of all recursive reals form a field under addition and multiplication [14]. Similarly, we can show that all $\emptyset'$-recursive reals form a strictly bigger but yet still countable field. Note that the relation R on recursive Cauchy sequences such that $R(\{r_n\}, \{s_n\})$ if and only if they converge to the same limit form an equivalent class. Let RS be the set of all equivalent classes under R on recursive Cauchy sequences, and we have

Corollary 10. *The field of all $\emptyset'$-recursive reals is isomorphic to RS.*

The following are some common operations in elementary analysis and their relationship to $\emptyset'$-recursiveness. It is known that the integral of a recursive function on a bounded interval must also be recursive [13], and hence the definite integral over a recursive interval must be a recursive real. However that is not necessarily true on an unbounded domain. In fact, we have

Theorem 11. *A real number a is $\emptyset'$-recursive if and only if there exists a recursive function f on $\mathbb{R}$ such that the definite integral $\int_{-\infty}^{\infty} f(t)\,\mathrm{d}t = a$.*

Proof. ($\Rightarrow$) Suppose a is $\emptyset'$-recursive. By Theorem 9, there exists a recursive sequence $\{r_n\}$ converging to a. Define the difference sequence $\{q_n\}$ as follow: $q_1=r_1$ and $q_n=r_n-r_{n-1}$ for $n>1$. So we have $\sum q_n=\lim r_n=a$. Now define the recursive function f on $\mathbb{R}$. Let $f(t)=0$ for $t<0$ and on every positive interval $[n,n+1]$, f is a triangle such that $f(n)=0=f(n+1)$, $f(n+\frac{1}{2})=2\cdot q_n$ and interpolate linearly in between. So we have

$$\int_{-\infty}^{\infty} f(t)\,\mathrm{d}t=\int_{0}^{\infty} f(t)\,\mathrm{d}t=\sum_{n=0}^{\infty}\int_{n}^{n+1} f(t)\,\mathrm{d}t=\sum_{n=0}^{\infty} q_n=\lim_{n\to\infty} r_n=a.$$

($\Leftarrow$) Suppose f on $\mathbb{R}$ is recursive and $\int_{-\infty}^{\infty} f(t)\,\mathrm{d}t=a$ is finite. Since the definite integral of a recursive function on a finite interval $[-n,n]$ is a recursive real, we can approximate it by a rational r_n within an error of 2^{-n}. Note that this approximation can be done uniformly in n. Now we have a recursive sequence of rationals $\{r_n\}$ converging to a, and thus a is $\emptyset'$-recursive. $\square$

There is a similar observation about the derivatives of a recursive function at recursive points.

Theorem 12. *A real number a is $\emptyset'$-recursive if and only if there exists a recursive function f on $[-1,1]$ such that f is differentiable at* 0 *and $f'(0)=a$.*

Proof. ($\Rightarrow$) Suppose a is $\emptyset'$-recursive. Theorem 9 shows that there exists a recursive sequence $\{r_n\}$ converging to a. We define a recursive sequence of functions on $[-1,1]$ as follows:
Stage $s=0$. Let $n_0=1$, $f_0(0)=0$ and $f_0(\pm 1)=\pm r_1$.[5] Interpolate linearly in between.
Stage $s+1$. Let n_{s+1} be the least n such that $n>n_s$ and $|r_n/n|\leqslant 2^{-(s+1)}$. We known that n_{s+1} exists because r_n converges. Define $f_{s+1}(0)=0$ and $f_{s+1}(\pm 1/n_m)=\pm r_{n_m}/n_m$ for $m=0,\dots,s+1$. Interpolate linearly in between. End of construction.

It is clear from the construction that $\{f_s\}$ is recursive. Now we show that $\{f_s\}$ also converges uniformly and effectively. Let $p>q\geqslant s+1$. We have $f_p(x)=f_q(x)$ for $|x|\geqslant 1/n_q$. For $|x|<1/n_q$, we have

$$|f_q(x)-f_p(x)|\leqslant|f_q(x)|+|f_p(x)|\leqslant 2^{-q}+2^{-q}=2^{-q+1}\leqslant 2^{-s}.$$

Hence $f=\lim f_s$ is recursive by Theorem 4. We have

$$f'_{+}(0)=\lim_{x\to 0^+}\frac{f(x)}{x}=\lim_{s\to\infty}\frac{f(1/n_s)}{1/n_s}=\lim_{s\to\infty}\frac{f_s(1/n_s)}{1/n_s}=\lim_{s\to\infty} r_{n_s}=a.$$

The second equality in the above holds because for all $x>0$, there exists an s such that $1/n_{s+1}\leqslant x<1/n_s$ and $f(x)$ in this region is a linear combination of $f(1/n_{s+1})$ and $f(1/n_s)$. Therefore $f(x)/x$ can be expressed as an expression sandwiched between two such linear combinations in terms of s with the same limit as $s\to\infty$, a consequence when $x\to 0^+$.

[5] The notation $f(\pm x)=\pm y$ means $f(+x)=+y$ and $f(-x)=-y$.

Similarly, $f'_{-}(0)=a$, and therefore f is differentiable at 0, and has value a.

($\Leftarrow$) Suppose f is differentiable at 0, and $f'(0)=a$. Since $a_n = n\cdot f(1/n)$ is recursive, we can approximate a_n by a rational r_n within an error of 2^{-n}. Furthermore, this can be done uniformly in n. Clearly $\{r_n\}$ converges to a. Hence a is $\emptyset'$-recursive. $\square$

This theorem expands Myhill's theorem ([12], also in [13]) which says there exists a recursive and continuously differentiable function whose derivative is not recursive, in the sense that it states this derivative cannot be an arbitrary non-recursive real, but has to be a $\emptyset'$-recursive one.

4. Relatively recursive real functions

Now we can turn into the definition of a relatively recursive real function:

Definition. A function $f:\mathbb{R}\to\mathbb{R}$ is *B-recursive* if there exists a two-oracle Turing machine M such that $(\forall x\in\mathbb{R})(\forall\phi\in CF_x)(\forall n\in\mathbb{N})\ |M^{B,\phi}(n)-f(x)|\leqslant 2^{-n}$. We say that f is B-recursive on an interval $[0,1]$ if the above inequality is true for all $x\in[0,1]$.

The idea is the same as a recursive function but now we can get "help" from the oracle B. The above definition is a natural extension of a B-recursive real and a recursive function. As usual, we will be mostly interested in a $\emptyset'$-recursive function. In the following, we define the notion of a "limiting recursive" function which can be proved to be equivalent to a $\emptyset'$-recursive function on a bounded interval. It is more useful in some setting because it is stated in terms of limit and convergence.

Definition. A function $f:[0,1]\to\mathbb{R}$ is *uniformly limiting recursive* if there exists a recursive sequence of functions $\{f_n\}$ on $[0,1]$ such that f_n converges uniformly to f on $[0,1]$.

In other words, f is uniformly limiting recursive if there exists an oracle Turing machine M such that $M(n,\cdot)$ computes f_n, i.e.,

$$(\forall x\in[0,1])(\forall\phi\in CF_x)(\forall n,p\in\mathbb{N})\,[|M^{\phi}(n,p)-f_n(x)|\leqslant 2^{-p}]$$

and a modulus function $d:\mathbb{N}\to\mathbb{N}$ for the uniform convergence, i.e., if $n\geqslant d(p)$, then $|f_n(x)-f(x)|\leqslant 2^{-p}$ for all $x\in[0,1]$. Note that we do not require the modulus to be recursive relative to any oracle B. It turns out that it is $\emptyset'$-recursive.

To show the implication of one direction, we need to first show that the modulus function for the uniform convergence in a uniformly limiting recursive is recursive in $\emptyset'$.

Lemma 13. *If f is uniformly limiting recursive on* $[0,1]$, *then the modulus for the uniform convergence is recursive in* $\emptyset'$.

Proof. Let $\{f_n\}$ be recursive and converges uniformly to f. Define

$$d_t(p) = (\mu n)\{(\forall x \in [0,1])(\forall i,j \in \{n,\dots,n+t\}) \mid f_i(x) - f_j(x)| \leqslant 2^{-p})\}.$$

Let $d(p) = \lim_{t\to\infty} d_t(p)$. We know that the limit exists because f_n converges uniformly and d is the required modulus. If we can show that d_t is recursive, then by the Limit Lemma 8, d is $\emptyset'$-recursive and we are done.

Let M be the Turing machine computing the sequence of functions $\{f_n\}$. Fix integers p, n and t. For each x in $[0,1]$, consider the computations $M^{\beta_x}(i, p+3)$ for $i \in \{n,\dots,n+t\}$ (recall that $\beta_x \in CF_x$ is the standard Cauchy function for x). For each i, there is a maximum number of terms used on β_x. Let k_x be the maximum among the number of terms used on β_x on all $i \in \{n,\dots,n+t\}$. Let $c_x = \beta_x(k_x)$, $l_x = c_x - 2^{-k_x}$ and $r_x = c_x + 2^{-k_x}$ where $\beta_x(k_x)$ denotes the k_x-th term of β_x. Now $\forall y \in (l_x, r_x)$, there is a $\psi \in CF_y$ such that ψ and β_x are the same on the first k_x terms. Thus for any $i \in \{n,\dots,n+t\}$, we have $M^{\beta_x}(i,p+3) = M^{\psi}(i,p+3)$ since the computations never use more than k_x terms on the oracle. Therefore, we have $\forall y \in (l_x, r_x)$ and $\forall i \in \{n,\dots,n+t\}$,

$$\begin{aligned} |f_i(x) - f_i(y)| &\leqslant |f_i(x) - M^{\beta_x}(i,p+3)| + |M^{\psi}(i,p+3) - f_i(y)| \\ &\leqslant 2^{-(p+2)}. \end{aligned} \tag{1}$$

Now $\{(l_x, r_x)\colon x \in [0,1]\}$ is an open cover for $[0,1]$. By the Heine–Borel Theorem, there is a finite subcover $\{(l_{x_s}, r_{x_s})\}_{s=1}^{u}$ for $[0,1]$. We will describe an algorithm below to find one such subcover. Note that for $s = 1,\dots,u$, the number $c_{x_s} = \beta_{x_s}(k_{x_s})$ is a dyadic rational. Let $\phi_s = \beta_{c_{x_s}}$ be the Cauchy function for c_{x_s}. This ϕ_s and β_{x_s} have the same initial k_{x_s} terms and hence $l_{c_{x_s}} = l_{x_s}$ and $r_{c_{x_s}} = r_{x_s}$. This suggests that we can work on the dyadic rationals $\mathbb{D}$ instead of arbitrary reals. We use the following algorithm to compute $d_t(p)$:

for $n = 1$ **to** ∞ **do** {
  for $q = 1$ **to** ∞ **do** {
    if $\bigcup_{x \in \mathbb{D}_q^{[0,1]}} (l_x, r_x)$ is a cover for $[0,1]$
    then break
  }
  let $\{(l_{x_s}, r_{x_s})\}_{s=1}^{u}$ covers $[0,1]$
  if $\forall i,j \in \{n,\dots,n+t\}$ **and** $\forall s \in \{1,\dots,u\}$,
    $|M^{\phi_s}(i,p+4) - M^{\phi_s}(j,p+4)| \leqslant 2^{-(p+2)}$
  then halt and output $d_t(p) = n$.
}

The inner for-loop will halt for each n, t, p because there exists a finite cover of $[0,1]$ with dyadic rational as center points of the intervals as discussed. The outer for-loop will halt because $\{f_n\}$ is uniformly Cauchy, i.e., if i and j are big enough,

$|f_i(z) - f_j(z)| \leqslant 2^{-(p+3)}$ for all $z \in [0,1]$. Hence

$$\begin{aligned}
&|M^{\phi_s}(i, p+4) - M^{\phi_s}(j, p+4)| \\
&\quad \leqslant |M^{\phi_s}(i, p+4) - f_i(c_{x_s})| + |f_i(c_{x_s}) - f_j(c_{x_s})| + |f_j(c_{x_s}) - M^{\phi_s}(j, p+4)| \\
&\quad \leqslant 2^{-(p+4)} + 2^{-(p+3)} + 2^{-(p+4)} \\
&\quad = 2^{-(p+2)}
\end{aligned}$$

and therefore the above algorithm always halts and so d_t is a recursive function. It remains to check that d_t does what we want. Let $n = d_t(p)$ and $\{(l_{x_s}, r_{x_s})\}_{s=1}^{u}$ be the cover of $[0,1]$ found in the algorithm. For each $x \in [0,1]$, there is an s such that $x \in (l_{x_s}, r_{x_s})$. Now for all $i, j \in \{n, \ldots, n+t\}$,

$$\begin{aligned}
|f_i(x) - f_j(x)| &\leqslant |f_i(x) - f_i(c_{x_s})| + |f_i(c_{x_s}) - M^{\phi_s}(i, p+4)| \\
&\quad + |M^{\phi_s}(i, p+4) - M^{\phi_s}(j, p+4)| \\
&\quad + |M^{\phi_s}(j, p+4) - f_j(c_{x_s})| + |f_j(c_{x_s}) - f_j(x)| \\
&= A + B + C + D + E
\end{aligned}$$

where

$$\begin{aligned}
&A \leqslant 2^{-(p+2)} && \text{by inequality (1)}, \\
&B \leqslant 2^{-(p+4)} && \text{by definition of } M, \\
&C \leqslant 2^{-(p+2)} && \text{by construction of } d_t(p), \\
&D \leqslant 2^{-(p+4)} && \text{by definition of } M, \\
&E \leqslant 2^{-(p+2)} && \text{by inequality (1)}.
\end{aligned}$$

Hence $A+B+C+D+E \leqslant 2^{-p}$. Note that the inequality does not depend on x, therefore we have $|f_i(x) - f_j(x)| \leqslant 2^{-p}s$ for all $x \in [0,1]$ and $i, j \in \{n, \ldots, n+t\}$. □

Theorem 14. *If f is uniformly limiting recursive on* $[0,1]$, *then f is $\emptyset'$-recursive on* $[0,1]$.

Proof. Suppose M is the oracle Turing machine that computes the recursive sequence of functions $\{f_n\}$ converging uniformly to f. Let d be the modulus for the uniform convergence. By the previous lemma, d is recursive in $\emptyset'$. Now define the oracle Turing machine M_1 by $M_1^{K,\phi}(n) = M^{\phi}(d(n+1), n+1)$ for all n and all ϕ. Note that M_1 is recursive in $\emptyset'$ because M is recursive and d is recursive in $\emptyset'$. Now M_1 computes f because $(\forall x \in [0,1])(\forall \phi \in CF_x)(\forall n \in \mathbb{N})$, we have

$$\begin{aligned}
&|M_1^{K,\phi}(n) - f(x)| = |M^{\phi}(d(n+1), n+1) - f(x)| \\
&\quad \leqslant |M^{\phi}(d(n+1), n+1) - f_{d(n+1)}(x)| + |f_{d(n+1)}(x) - f(x)| \\
&\quad \leqslant 2^{-(n+1)} + 2^{-(n+1)} = 2^{-n}.
\end{aligned}$$

Hence f is $\emptyset'$-recursive on $[0,1]$. □

To show the converse direction is more difficult; we need to use a finite injury priority argument. First we need a lemma to show that a $\emptyset'$-recursive function is uniformly continuous on a compact domain.

Lemma 15. *If f is $\emptyset'$-recursive on* $[0,1]$, *then f is uniformly continuous on* $[0,1]$ *and the modulus for the uniform continuity is recursive in $\emptyset'$.*

The lemma generalizes a classical result of Grzegorczyk [2] in which he showed that f is recursive on $[0,1]$ implies it is uniformly effectively continuous there with a recursive modulus. A proof of this found in [9, Theorem 2.13] can be applied to our lemma here with only a minor modification of adding an oracle K and hence we will omit it. In fact, we can substitute the oracle K with any other oracle B without affecting the proof. The only difference is that we will get a modulus function that is recursive in B.

Theorem 16. *A function f is $\emptyset'$-recursive on* $[0,1]$ *if and only if f is uniformly limiting recursive on* $[0,1]$.

Proof. ($\Leftarrow$) By Theorem 14.

($\Rightarrow$) Let M be the Turing machine that computes f:

$$(\forall x \in [0,1])(\forall \phi \in CF_x)(\forall e \in \mathbb{N})\,[|M^{K,\phi}(e) - f(x)| \leqslant 2^{-e}],$$

where K is the halting set oracle. Let $\{K_s\}$ be an enumeration of K as discussed before and let the notation M_s denote the Turing machine M running for s steps. By the previous lemma, f is uniformly continuous on $[0,1]$ and has a modulus m which is recursive in $\emptyset'$. Without loss of generality, we can assume that m is non-decreasing. So we have

$$(\forall x, y \in [0,1])\,[|x - y| \leqslant 2^{-m(e)} \Rightarrow |f(x) - f(y)| \leqslant 2^{-e}]$$

and

$$(\forall e, e' \in \mathbb{N})\,[e < e' \Rightarrow m(e) \leqslant m(e')].$$

Since $m \leqslant_T \emptyset'$, there exists a recursive sequence of functions (on integers) $\{m_s\}$ such that $\lim_s m_s = m$ by the Limit Lemma 8.

To show that f is uniformly limiting recursive, we use a finite injury priority argument to construct a recursive sequence $\{f_s\}$ such that it satisfies all the *requirements* R_e, $e \geqslant 1$:

$$R_e : (\exists s_e)(\forall s \geqslant s_e)(\forall x \in [0,1])\,[|f_s(x) - f(x)| \leqslant 2^{-(e-4)}].$$

Clearly if $\{f_s\}$ can satisfy all requirements R_e, then it converges to f uniformly and we are done.

Construction of $\{f_s\}$:
Stage $s=0$. Let $f_0 \equiv 0$ on $[0,1]$.
Stage $s+1$. For all $e \leqslant s+1$, if

$$m_{s+1}(e) \neq m_s(e)$$

or

$$(\exists x \in \mathbb{D}^{[0,1]}_{m_{s+1}(e)})\,[M^{K_{s+1},\beta_x}_{s+1}(e) \neq M^{K_s,\beta_x}_s(e)]$$

(there is at least one dyadic rational in the set mentioned such that the value of M has changed since last stage), then R_e and all R_i for $i>e$ are said to be *injured* and *require attention.*

Choose least $e \leqslant s+1$ such that R_e requires attention. If $e=1$ and

$$(\forall x \in \mathbb{D}^{[0,1]}_{m_{s+1}(1)})\,[M^{K_{s+1},\beta_x}_{s+1}(1)\downarrow],$$

define f_{s+1} to be a piecewise linear function on $[0,1]$ with turning points on $x \in \mathbb{D}^{[0,1]}_{m_{s+1}(1)}$ and values $f_{s+1}(x) = M^{K_{s+1},\beta_x}_{s+1}(1)$. The requirement R_1 is said to have *received attention* and let $s_1 = s+1$ be the *stage when* R_1 *receives attention.*

If $e>1$ (this implies implicitly that all $R_{e'}$, $e'<e$ have received attention), and we have

$$(\forall x \in \mathbb{D}^{[0,1]}_{m_{s+1}(e)})\,[M^{K_{s+1},\beta_x}_{s+1}(e)\downarrow],$$

$$m_{s+1}(e) \geqslant m_{s+1}(e-1)$$

and

$$(\forall x \in \mathbb{D}^{[0,1]}_{m_{s+1}(e)})\,[|M^{K_{s+1},\beta_x}_{s+1}(e) - f_{s_{e-1}}(x)| \leqslant 2^{-(e-3)}]$$

then let f_{s+1} be a piecewise linear function on $[0,1]$ with turning points on $x \in \mathbb{D}^{[0,1]}_{m_{s+1}(e)}$ and $f_{s+1}(x) = M^{K_{s+1},\beta_x}_{s+1}(e)$. In this case, we say that R_e has *received attention* and let $s_e = s+1$ be *the stage when* R_e *receives attention.*

If no such e exists, or R_e does not receive attention in the above process, let $f_{s+1} = f_s$.

Note that s_e will change if R_e is injured but receives attention again in a later stage. However, this can happen only finitely often:

Claim. *For each e, R_e will eventually receive attention and never get injured later.*

Proof. By induction on e:

Case $e=1$: Let s be a stage such that $m_t(1) = m(1)$ for all $t \geqslant s$ (m stabilizes after stage s). Let s' be a stage such that for all $t > s'$

$$(\forall x \in \mathbb{D}^{[0,1]}_{m(1)})\,[M^{K_t,\beta_x}_t(1) = M^{K,\beta_x}(1)].$$

If R_1 has not received attention by stage $\max(s,s')$, it will require attention and receive it and never get injured later because all computations involved have stabilized.

Case $e>1$: By induction hypothesis, suppose that all $R_{e'}$ where $e'<e$ have received attention and never get injured after stage u. This implies that all computations involving $R_{e'}$ have stabilized. Let v be a stage such that $m_t(e)=m(e)$ for all $t \geqslant v$ and let w be a stage such that for all $t>w$

$$(\forall x \in \mathbb{D}^{[0,1]}_{m(e)})\,[M^{K_t,\beta_x}_t(e)=M^{K,\beta_x}(e)].$$

Let $s=\max(u,v,w)$. Hence

$$m_s(e)=m(e) \geqslant m(e-1)=m_s(e-1)$$

and therefore $\mathbb{D}^{[0,1]}_{m_s(e-1)} \subset \mathbb{D}^{[0,1]}_{m_s(e)}$. Also, since $f_{s_{e-1}}$ is piecewise linear on $\mathbb{D}^{[0,1]}_{m_s(e-1)}$, for any $x \in \mathbb{D}^{[0,1]}_{m_s(e)}$, there is a $d \in \mathbb{D}^{[0,1]}_{m_s(e-1)}$ such that

$$|x-d| \leqslant 2^{-m_s(e-1)}$$

and

$$|M^{K_s,\beta_x}_s(e)-f_{s_{e-1}}(x)| \leqslant |M^{K_s,\beta_x}_s(e)-f_{s_{e-1}}(d)|.$$

Therefore, for $s=\max(u,v,w)$,

$$\begin{aligned}
&|M^{K_s,\beta_x}_s(e)-f_{s_{e-1}}(x)|\\
&\quad \leqslant |M^{K_s,\beta_x}_s(e)-f_{s_{e-1}}(d)|\\
&\quad \leqslant |M^{K_s,\beta_x}_s(e)-f(x)|+|f(x)-f(d)|+|f(d)-f_{s_{e-1}}(d)|\\
&\quad = |M^{K,\beta_x}(e)-f(x)|+|f(x)-f(d)|+|f(d)-M^{K,\beta_d}(e-1)|\\
&\qquad\quad \text{since all computations involving } R_{e-1} \text{ have stabilized}\\
&\quad \leqslant 2^{-e}+|f(x)-f(d)|+2^{-(e-1)}\\
&\qquad\quad \text{by definition of } M\\
&\quad \leqslant 2^{-e}+2^{-(e-1)}+2^{-(e-1)}\\
&\qquad\quad \text{since } m_s(e-1)=m(e-1) \text{ and } |x-d| \leqslant 2^{-m(e-1)}\\
&\quad \leqslant 3\cdot 2^{-(e-1)}\\
&\quad \leqslant 2^{-(e-3)}.
\end{aligned}$$

If R_e has not received attention by this stage s, it will require attention and receive it. R_e will not be injured again because all computations have stabilized and there is no higher priority requirements to injury it. □

Now we have to show that if all R_e have received attention, then all R_e are satisfied.

Claim. *For all e, the requirement R_e is satisfied.*

Proof. Fix e, we need to show that there exists a stage s_e such that for all $s \geqslant s_e$ and for all $x \in [0,1]$, we have $|f_s(x) - f(x)| \leqslant 2^{-(e-4)}$. Let s_e be the stage that the requirement R_e receives attention for the last time and never gets injured in later stages. Let s be any stage greater than s_e. Without loss of generality, we can assume that s is a stage where a requirement R_n, $n > e$ receives attention, because otherwise if s' is the last stage before s such that a requirement receives attention, then, by construction, $f_s = f_{s'}$.

Let $R_e, R_{e+1}, \ldots, R_n$ be all the requirements that have received attentions and not injured for the last time *from stages s_e to s*. Let s_i be the stage that R_i receives attention for $i = e, \ldots, n$. So, we have $s = s_n$. Note that $R_{e+1}, \ldots, R_n$ may be injured at stages later than s but that does not matter. Consider stages s_i to s_{i+1} for $e \leqslant i < n$, we have

$$\begin{aligned} m_s(i) &= m_{s_{i+1}}(i) && \text{since } R_i \text{ is not injured through stage } s \\ &\leqslant m_{s_{i+1}}(i+1) && \text{since } R_{i+1} \text{ receives attention at stage } s_{i+1} \\ &= m_s(i+1) && \text{since } R_{i+1} \text{ is not injured through stage } s, \end{aligned}$$

and thus $\mathbb{D}^{[0,1]}_{m_s(i)} \subset \mathbb{D}^{[0,1]}_{m_s(i+1)}$. Since $f_{s_{i+1}}$ and f_{s_i} are piecewise linear on $\mathbb{D}^{[0,1]}_{m_s(i+1)}$, for any fixed $x \in [0,1]$, there is a $d \in \mathbb{D}^{[0,1]}_{m_s(i+1)}$ such that

$$|x - d| \leqslant 2^{-m_s(i+1)}$$

and

$$|f_{s_{i+1}}(x) - f_{s_i}(x)| \leqslant |f_{s_{i+1}}(d) - f_{s_i}(d)|.$$

Hence

$$\begin{aligned} |f_{s_{i+1}}(x) - f_{s_i}(x)| &\leqslant |f_{s_{i+1}}(d) - f_{s_i}(d)| \\ &= |M^{K_{s_{i+1}}, \beta_d}_{s_{i+1}}(i+1) - f_{s_i}(d)| && \text{by definition of } f_{s_{i+1}} \\ &\leqslant 2^{-(i-2)} && \text{by construction of } f_{s_{i+1}}. \end{aligned}$$

Also, for a fixed $x \in [0,1]$, there exists $d, d' \in \mathbb{D}^{[0,1]}_{m(e)}$ such that $d \leqslant x \leqslant d'$ and $d' - d = 2^{-m(e)}$. Now

$$
\begin{aligned}
&|f(x) - f_{s_e}(x)| \\
&\quad \leqslant |f(x) - f(d)| + |f(d) - f_{s_e}(d)| + |f_{s_e}(d) - f_{s_e}(x)| \\
&\quad \leqslant |f(x) - f(d)| + |f(d) - f_{s_e}(d)| + |f_{s_e}(d) - f_{s_e}(d')| \\
&\qquad \text{since } f_{s_e} \text{ is piecewise linear on } \mathbb{D}_{m(e)}^{[0,1]} \\
&\quad \leqslant |f(x) - f(d)| + |f(d) - f_{s_e}(d)| \\
&\qquad + |f_{s_e}(d) - f(d)| + |f(d) - f(d')| + |f(d') - f_{s_e}(d')| \\
&\quad \leqslant 2^{-e} + 2 \cdot |f(d) - f_{s_e}(d)| + 2^{-e} + |f(d') - f_{s_e}(d')| \\
&\qquad \text{since } m(e) \text{is the modulus function for } f \\
&\quad = 2 \cdot 2^{-e} + 2 \cdot |f(d) - M_{s_e}^{K_{s_e}, \beta_d}(e)| + |f(d') - M_{s_e}^{K_{s_e}, \beta_{d'}}(e)| \\
&\qquad \text{by construction of } f_{s_e} \\
&\quad = 2 \cdot 2^{-e} + 2 \cdot |f(d) - M^{K, \beta_d}(e)| + |f(d') - M^{K, \beta_{d'}}(e)| \\
&\qquad \text{since } R_e \text{ never gets injured after stage } s_e \\
&\quad \leqslant 2 \cdot 2^{-e} + 2 \cdot 2^{-e} + 2^{-e} \\
&\qquad \text{by definition of } f \text{ and } M \\
&\quad = 5 \cdot 2^{-e} \\
&\quad \leqslant 2^{-(e-3)}.
\end{aligned}
$$

Combining the above analysis, we have for any $x \in [0, 1]$,

$$
\begin{aligned}
|f(x) - f_s(x)| &= |f(x) - f_{s_n}(x)| \\
&\leqslant |f(x) - f_{s_e}(x)| + \sum_{i=e}^{n-1} |f_{s_i}(x) - f_{s_{i+1}}(x)| \\
&\leqslant 2^{-(e-3)} + \sum_{i=e}^{n-1} 2^{-(i-2)} \\
&\leqslant 2^{-(e-3)} + 2^{-(e-3)} \\
&= 2^{-(e-4)}
\end{aligned}
$$

and hence requirement R_e is satisfied. □

Corollary 17 (Weierstrass approximation Theorem in $\emptyset'$). *A function f on $[0, 1]$ is $\emptyset'$-recursive if and only if there exists a recursive sequence of polynomials $\{p_n\}$ on $[0, 1]$ such that p_n converges uniformly to f.*

Proof. ($\Leftarrow$) By Theorem 16.

($\Rightarrow$) The proof of the Effective Weierstrass Approximation Theorem (Theorem 5) is constructive, i.e., for a recursive function on $[0, 1]$, we can effectively find a recursive sequence of polynomials converging uniformly to it. Suppose f is $\emptyset'$-recursive, for each piecewise linear recursive function f_s constructed in the proof of Theorem 16, we can thus effectively find a recursive polynomials p_s such that $|f_s(x) - p_s(x)| \leqslant 2^{-s}$ for all $x \in [0,1]$. Hence, $\{p_s\}$ is a recursive sequence of polynomials, and it is clear that p_s tends to f uniformly. □

The third characterization of $\emptyset'$-recursive functions is a natural extension to Theorem 3.

Theorem 18. *A function f on $[0,1]$ is $\emptyset'$-recursive if and only if*
(1) *for any recursive enumeration $\{r_n\}$ of all rationals in $[0,1]$, $\{f(r_n)\}$ is a $\emptyset'$-recursive sequence of reals, and,*
(2) *f is uniformly continuous on $[0,1]$ with a $\emptyset'$-recursive modulus for the uniform continuity.*

Proof. ($\Rightarrow$) (2) follows from Lemma 15. Suppose M is the Turing machine computing f, and M_1 enumerates $\{r_n\}$, i.e.,

$$(\forall x \in [0,1])(\forall \phi \in CF_x)(\forall m \in \mathbb{N})\,[|M^{K,\phi}(m) - f(x)| \leqslant 2^{-m}]$$

and

$$(\forall n \in \mathbb{N})\,[M_1(n) = r_n].$$

Define $M_2^K(n,m) = M^{K,\beta_{r_n}}(m)$. Note that the terms in β_{r_n} can be computed recursively and we need to compute only those terms that are required in the evaluation. Hence

$$|M_2^K(n,m) - f(r_n)| = |M^{K,\beta_{r_n}}(m) - f(r_n)| \leqslant 2^{-m}$$

and $\{f(r_n)\}$ is a $\emptyset'$-recursive sequence of reals.

($\Leftarrow$) Let d be a $\emptyset'$-recursive modulus for the uniform continuity of f. Let $\{r_n\}$ be a recursive enumeration of rationals in $[0,1]$, and so by (1), there exists M_1 such that

$$|M_1^K(n,m) - f(r_n)| \leqslant 2^{-m}.$$

Define $M^{K,\phi}(n) = M_1^K(i, n+1)$ where i is the least j such that $r_j = \phi(d(n+1))$ (the $d(n+1)$-th term of ϕ.) Note that such i exists and the computation for i is recursive in K. To show that M computes f, we have for all $x \in [0,1]$, for all $\phi \in CF_x$,

$$\begin{aligned}
|M^{K,\phi}(n) - f(x)| &= |M_1^K(i,n+1) - f(x)| \\
&\leqslant |M_1^K(i,n+1) - f(r_i)| + |f(r_i) - f(x)| \\
&= |M_1^K(i,n+1) - f(r_i)| + |f(\phi(d(n+1))) - f(x)| \\
&\leqslant 2^{-(n+1)} + 2^{-(n+1)} \\
&= 2^{-n}.
\end{aligned}$$

The inequality in the fourth line holds because, by definition, $|\phi(d(n+1)) - x| \leqslant 2^{-d(n+1)}$ and since d is the modulus, $|f(\phi(d(n+1))) - f(x)| \leqslant 2^{-(n+1)}$. □

Theorems 16 and 18 show that all three characterizations of a $\emptyset'$-recursive functions are equivalent. Since they are stated in quite different terms, this leads us to believe that the definition of a $\emptyset'$-recursive function is a correct and natural one. With this definition, we are now able to classify functions that were previously only proven to be non-recursive.

In the topics of differentiation and integration, it has been shown that the derivative of a continuously differentiable recursive function is not necessarily recursive [12]. But if f is twice continuously differentiable (actually the weaker condition that f' has a recursive modulus of uniform continuity suffices), then it can be shown that f' is recursive [13]. Here we show that although the derivative of a continuously differentiable recursive function may not be recursive, it cannot have an arbitrary high degree of unsolvability.

Theorem 19. *If f is recursive and continuously differentiable on $[0,1]$ (i.e., $f \in C^1[0,1]$), then f' is $\emptyset'$-recursive on $[0,1]$.*

Proof. Let $f_n(x) = n \cdot (f(x+1/n) - f(x))$ be defined on $[0,\frac{1}{2}]$ for $n \geqslant 2$. It is clear that this sequence $\{f_n\}$ is recursive and f_n converges pointwise to f' on $[0,\frac{1}{2}]$. If we can show that the convergence is in fact uniform, then by Theorem 16, f' is $\emptyset'$-recursive on $[0,\frac{1}{2}]$. Using a similar technique, we can show that f' is also $\emptyset'$-recursive on $[\frac{1}{2},1]$. Hence f' must be $\emptyset'$-recursive on the entire interval $[0,1]$.

To show that f_n in fact converges uniformly to f' on $[0,\frac{1}{2}]$, fix $\varepsilon > 0$. Since f' is uniformly continuous on $[0,1]$, there exists an $\delta > 0$ such that for all $x, y \in [0,1]$, $|x-y| \leqslant \delta$ implies $|f'(x) - f'(y)| \leqslant \varepsilon$. Also, by the Mean Value Theorem, if $x \leqslant \frac{1}{2}$ and $n \geqslant 2$, then there exists a point $z \in (x, x+1/n)$ such that $f'(z) = n \cdot (f(x+\frac{1}{n}) - f(x)) = f_n(x)$. Let $\delta' = \min\{\delta, \frac{1}{2}\}$ and N be the least integer such that $1/N \leqslant \delta'$.

Now for all $n > N$, $1/n < 1/N \leqslant \delta'$ and hence for all $x \in [0,\frac{1}{2}]$, we have

$$\begin{aligned} |f_n(x) - f'(x)| &= |f'(z) - f'(x)| && \text{for some } z \in (x, x+\tfrac{1}{n}) \\ &\leqslant \varepsilon && \text{since } |z-x| < \tfrac{1}{n} \leqslant \delta' \leqslant \delta. \end{aligned}$$

Note that the choice of N above does not depend on x and hence the convergence is uniform. □

In contrast to the above result, integration does not necessarily bring an arbitrary $\emptyset'$-recursive function back into a recursive function. For instance, consider the function $f \equiv x$, where $x = \sum_{n \in K} 2^{-n}$, where K is the halting set. However, integration is indeed a nice operator in the sense that it "patches up" a lot of non-recursive functions. For illustration purpose, we state a theorem by Ko [9, Theorem 5.29], which gives a sufficient condition for the integral to be recursive:

Theorem 20 (Ko [9]). *Let f be bounded and recursively approximable on $[0,1]$. Then $F(x) = \int_0^x f(t)\,dt$ is recursive on $[0,1]$.*

We shall not give the precise definition of *recursively approximability* here as we do not need it. It suffices to say that it is a fairly large class which include some nowhere continuous functions. For further characteristic of a recursively approximable function, one can refer to [9].

5. Conclusion

We have defined relatively recursive functions in this paper. However, no matter which oracle we use, the resulting function is always continuous (Lemma 15). The intuitive reason is because a Turing machine computation is a finite process if it is going to halt on all inputs on a compact domain. Therefore it will not be able to differentiate points that are arbitrarily close together, and hence the function that it can approximate must be continuous. Suppose we are willing to relax the conditions on compact domain and/or halt-on-all-inputs, interesting results can arise. The class of recursively approximable functions of Ko is one such example. Ho in [7] has defined two other classes, the *almost everywhere recursive* functions and the *weakly almost everywhere recursive* functions, which lie strictly in between the classes of recursive functions and recursively approximable functions. They are in some sense better-behaved than the recursively approximable functions as they are more "structured". For example, the class of almost everywhere recursive functions captures the notion of step functions on bounded domains with recursive values and recursive breakpoints pretty well. Other interesting results include effective versions of the Lusin and Egoroff Theorems. Some open problems can be found there too.

Acknowledgement

Thanks to Stuart Kurtz for his many stimulating suggestions and ideas. Discussions with him have been very valuable to this paper.

References

[1] J. Caldwell, M.B. Pour-El, On a simple definition of computable function of a real variable – with applications to functions of a complex variable, Z. Math. Logik Grundlag. Math. 21 (1975) 1–19.
[2] A. Grzegorczyk, Computable functionals, Fundam. Math. 42 (1955) 168–202.
[3] A. Grzegorczyk, On the definitions of computable real continuous functions, Fundam. Math. 44 (1957) 61–71.
[4] A. Grzegorczyk, Some approaches to constructive analysis, in: A. Heyting (Ed.), Constructivity in Math., North-Holland, Amsterdam, 1959, pp. 43–61.
[5] C.K. Ho, Relatively recursive reals and real functions, Technical Report TR-94-02, Department of Computer Science, The University of Chicago, 1994.
[6] C.K. Ho, Convergence in recursive analysis, Ph.D. Thesis, TR-94-25, Department of Computer Science, The University of Chicago, 1994.
[7] C.K. Ho, Beyond recursive real functions, Inform. Computat. 124 (2) (1996) 113–126.
[8] K. Ko, Computational complexity of real functions, Theoret. Comput. Sci. 20 (1982) 323–352.

[9] K. Ko, Complexity Theory of Real Functions, Birkhäuser, Basel, 1991.
[10] S. Kurtz, Private Conversation, 1992.
[11] A. Mostowski, On computable sequences, Fundam. Math. 44 (1957) 37–51.
[12] J. Myhill, A recursive function, defined on a compact interval and having a continuous derivative that is not recursive, Michigan Math. J. 18 (1971) 97–98.
[13] M. Pour-El, J. Richards, Computability in Analysis and Physics, Springer, Berlin, 1988.
[14] H. Rice, Recursive real numbers, Proc. Amer. Math. Soc. 5 (1954) 784–791.
[15] H. Rogers, Jr., Theory of Recursive Functions and Effective Computability, McGraw-Hill, New York, 1967.
[16] W. Rudin, Real and Complex Analysis, 3rd ed., McGraw-Hill, New York, 1986.
[17] R. Soare, Recursively Enumerable Sets and Degrees, Springer, Berlin, 1987.
[18] E. Specker, Der Satz vom maximum in der rekursiven analysis, in: A. Heyting (Ed.), Constructivity in Math., North-Holland, Amsterdam, 1959, pp. 254–265.
[19] A. Turing, On computable numbers, with an application to the "Entscheidungsproblem", Proc. London Math. Soc. Ser. 2 42 (1936) 230–265; corr. 43 (1937) 544–546.